**Memorandum**

**Department of Transport and Main Roads** Enquiries

**Telephone +61 7**

**Facsimile** +61 7

Our ref

Your ref

Date 17 July 2018

To Miranda Blogg

Copy to CAVI Team

Subject Enablement of Media Agnostic Communications of I2V and I2I for non-time critical information

In an effort to ready local government, industry, and our customers for cooperative intelligent transport systems (C-ITS), the Queensland Department of Transport and Main Roads (TMR) has begun planning for a C-ITS pilot involving around 500 vehicles. One of the objectives of the pilot is to explore TMR’s obligations resulting from the adherence to (and the limitations of) the current C-ITS standards suite.

Transport jurisdictions are able to increase the value of C-ITS by sharing their information – lane and road closures, speed limits, on-going traffic incidents and so on – directly with the driver. This type of information is not time critical and hence, not reliant on the provision of short-range communications. The latency of 3G/4G is sufficient and in line with the agnostic model afforded by the C-ITS standards.

To date, international pilots have supported cellular transfer of messages using bespoke communications protocols on a case by case basis as per a particular pilot’s requirements. As such, each European pilot thus far has implemented its own (often bespoke, though standards based) communications standard. For example, ECO-AT uses a bespoke HTTP interface for cellular clients.

With enough development work, almost any protocol or combination thereof could be adapted to suit a C-ITS cellular environment. The focus of this analysis was to identify a protocol that –

* Suited a C-ITS environment “out of the box”;
* Was well established;
* Had an active support community;
* Had a good integration potential with new concepts, like IoT; and
* Met the majority of the pilot’s requirements and timeframes

This paper provides a “soft” analysis of the available protocol options for the broader international C-ITS community’s consideration – including ISO/TS 17429, TCP Encoded ETSI, HTTP, NTCIP, OCIT-C, MQTT, and OMG-DDS. ISO/TS 17429 is a peer to peer model intended for the transfer of 3G/4G messages between the cooperative stations (ITS-S). However, a number of limitations were identified –

* “off the shelf” products and source code are not available
* - it is bespoke to C-ITS, and therefore not used for other smart city use-cases
* none of the pilot ITS-S suppliers have used the standard at this time
* design, development, implementation and testing would be required by both the transport jurisdiction and the station providers
* add something about jurisdiction handover (movement of vehicles between states)??

Supporting most of the identified requirements (below), MQTT is a publish/subscribe style protocol, pushing and pulling all messages through a central message broker. It includes session and message management and is supported by the major cloud platform providers. The protocol is a “Hub and Spoke” design and does not adhere to the “Peer to Peer” model desirable for C-ITS, however, many of the centre to field use-cases (for example In Vehicle Signage, Reduced Speed Zone Warning, etc) are a by their nature a Hub and Spoke design and do not included a Peer to Peer component.

# TMR C-ITS pilot requirements

In late October of 2017, TMR hosted a communications workshop to both agree on a single network communications specification and protocol for the C-ITS pilot, and to present a protocol that posed an attractive option for consideration by the broader C-ITS community.

Included in the workshop were representatives from –

* Q-Free
* Cohda
* Kapsch
* Savari/Excel
* Transmax
* NetBI
* TMR
* WSP

Requirements of the protocol, mapped against the pilot’s objectives, are listed below.

| **Objective** | **C-ITS Pilot system must :** | **The protocol must:** |
| --- | --- | --- |
| **Support TMR readiness** | *explore TMRs obligations and limitations of C-ITS standards:* -include R-ITS-S and C-ITS-F  -generate a number of C-ITS message types  -include multi-vendors ITS-S products  -build a comms agnostic model to send/receive messages  -use a security credential management system | - meet C-ITS standards  - support peer to peer environment  - Include a process to find other C-ITS devices on a shared network  - use low (comparative) client resources  - offer automatic network/interface selection  - supports “persistent” messages  - has a low bandwidth overhead  -supports bi-directional, efficient messages transfer using 3G/4G |
| *explore impacts to TMRs systems and processes:*  -use or enhance system, services where appropriate -monitor and report on system's performance -identify and manage adverse impacts on the systems -use TMR staff to explore pilot system roles |  |
| *support TMR goal of a single integrated network:*  -integrate with other Queensland transport network operators |  |
| **Build Industry capability/ partnerships** | *share design and learning:*  -design can be used by others  -is consistent with national direction (C-ITS standards) -limit bespoke requirements to maximise interoperability -choose supported/ viable bespoke requirements where needed | - support hand-off with other Transport jurisdictions  - support “Hub n’ Spoke” model  - be scalable  has Cloud provider support  has Opensource community support  is Payload agnostic  - uses a known, well established protocol  - Is standardised rather than Bespoke (ISO Vs Bespoke)  - be readily adopted by vendors/simple implementation |
| *maximise industry capability and readiness:*  -include multiple vendors in the pilot -build a test bed for use by TMR and third parties -contribute findings to international C-ITS standards | -offers a possible contribution to standards |
| **Understand the safety benefits** | *measure safety impact of day 1 use cases:*  -include signals, hazards, and signage use-cases -include passenger vehicles  -use driver behaviour to infer a crash reduction -support baseline/ control and treatment periods data collection -support representative sample of participants -adopt naturalistic study  -collect validation data from external systems (BOM, cameras) |  |
| *get a sufficient quantity and quality of safety related data:*  -maximise participation (# of participants and interactions)  -maximise vehicle to infrastructure interactions  -maximise pilot duration -perform driver simulator studies when the data size is insufficient -harmonise use-case HMI, algorithms, and data collected -design the system to minimise system and device issues -collect /calculate system, device and data quality metrics -enable systemwide maintenance of devices | -supports efficient transfer of device data |
| *to meet ethical and legal requirements:*  -meet requirements of the QUT Ethics Committee and TMR Legal | - does not enable tracking of a vehicle |
| **Build public awareness and uptake** | *to build public awareness:*  -include large public participation; and some fleet participation |  |
| *to ensure a positive public perception:*  -messages that are accurate, timely, reliable, relevant  -ensure the HMI encourages a useful driver response -minimise participant workload (install, operate, maintain) -not cause damage to vehicles or TMR asset | -have low latency, message confirmation, and message resend for intermittent communications  -Supports message push rather than poll (Pub/Sub)  -Supports persistent messages  - Supports QoS mechanisms that ensures message priority and delivery |

# Options analysis

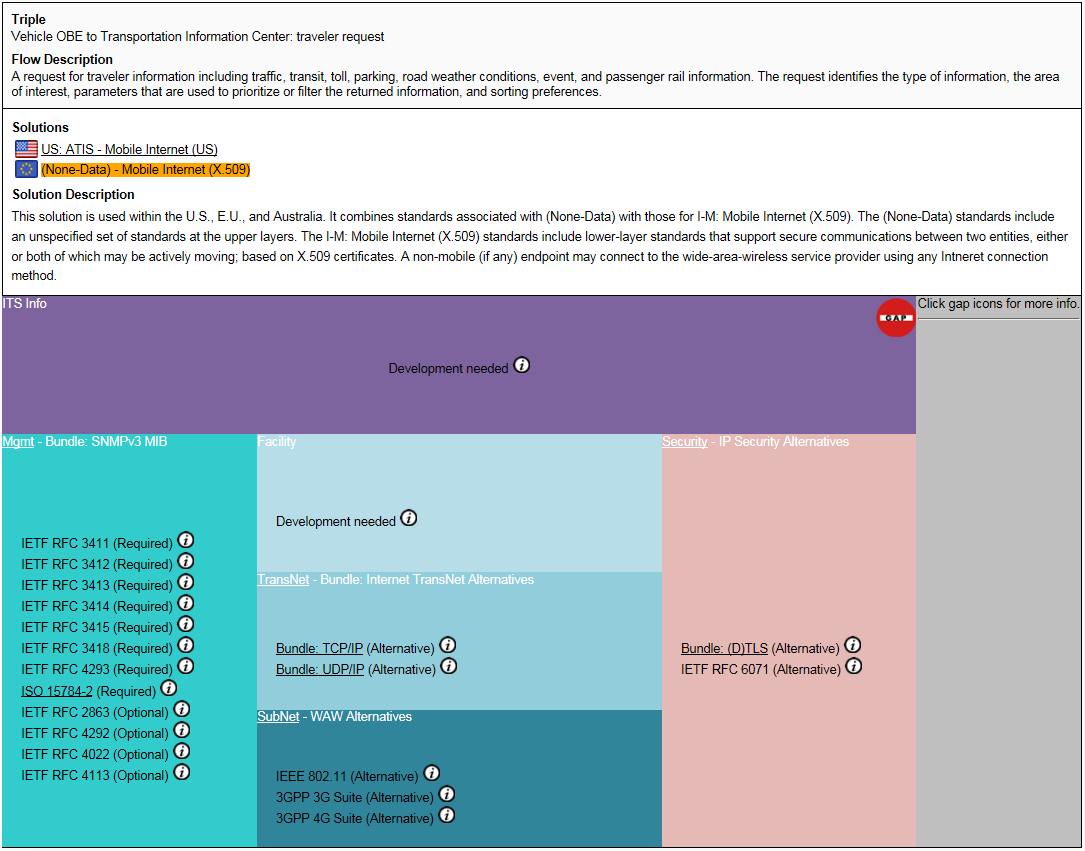
A number of protocols were offered as candidates during the workshop as outlined below. In addition, TMR investigated a number of IoT style protocols both prior to and post workshop that have been included for comparison purposes in the final choice of protocols for C-ITS.

The theoretical and mathematical backing against the protocol’s requirements was not performed due to;

* The qualitative rather than quantitative nature of the environmental factors and ;
* The dynamic nature and immaturity of the technology.

As such, this paper uses a qualitative SWOT analysis to distinguish between protocol choice for C-ITS and, with the consideration of the relatively new technology including references where available. Importantly, the time for planning, development and implementation in both the C-ITS-S and V-ITS-S for this protocol would be limited to 1 month total.

Efforts have been made by HGT7 to capture the standards and gaps for a number of use cases, for example; (<http://htg7.org/html/triples/s4d17f418.html>).



HTG7 – use case example for 3G/4G transfer of messages

However, as can be seen in the above example (and examined further below) that uses Cellular, TCP/IP and SNMP, it is 1) not specified in any ETSI standard, 2) appears to require further development and 3) potentially may not be the ideal collection of protocol choices for use in a new peer-to-peer ITS technology.

This paper follows on from the workshop and examines the following protocol choices –

* **TCP Encoded ETSI Message**
* **HTTP (SOAP, etc)**
* **NTCIP**
* **OCIT-C**
* **MQTT**
* **OMG-DDS**
* **ISO/TS 17429:2017**

**TCP Encoded ETSI Message**

TCP Encoded ETSI message is the simplest of the proposals explored. Encapsulating an ETSI defined packet in TCP (a common networking protocol) for communication between the transport jurisdiction and ITS-S.

|  |  |  |
| --- | --- | --- |
|  | **Helpful** | **Harmful** |
| **Inherent** | Strength   * Has a low bandwidth overhead * Includes transmission receipt (TCP packet ACK) | Weakness   * Transient communications require further design and development work * Unable to persist messages - Message, session and/or information management require design and development work * Unable to target geographic regions, countering its low bandwidth strength * No mechanism for automatic network/interface selection * Does not include a process to find other C-ITS devices on a shared network * No QoS mechanism |
| **Environment** | Opportunity   * Easiest and quickest protocol for station providers to implement – could be readily adopted by vendors | Threats   * Hub and Spoke model rather than Peer to Peer * Solution lifetime and roadmap questionable * Bespoke, not standards based |

Analysis:

Initially, this protocol appeared to have great advantages for TMR - relatively simple to implement and able to be achieved within the pilot’s timeframes. However, on closer examination the weaknesses inherent in this protocol choice was likely to require significant and bespoke design and development to overcome.

For example, there is no way for the transport jurisdiction to determine the network address of a station in order to send it a message. The station would have to report its address first, in order for the transport jurisdiction (or *any* station) to be able to send it information. This implies the development and maintenance of some sort of station and message tracking facility. These sorts of facilities are already part of a number of the other protocols examined and were largely seen as a duplication of effort by the decision making team.

Additionally, this choice does not adhere to any standard (local or international), reducing the likelihood of adoption by any further pilots, potentially leading to a complete rework. This was seen as undesirable by TMR.

Due to these combined factors, this protocol choice was not recommended.

**HTTP (SOAP, etc)**

Hypertext Transfer Protocol was originally introduced to standardise the transfer of “hypertext” information. Hypertext was popularised by its use in the development of the World Wide Web (www) and was primarily used as an easy way of presenting, navigating and linking documents across various sources.

This protocol (HTTP) has been extended over years to fulfil a variety of roles including –

* Transfer of large data files
* Machine-to-Machine transfer of data records
* Media/Video playback & “streaming”
* And more

|  |  |  |
| --- | --- | --- |
|  | **Helpful** | **Harmful** |
| **Inherent** | Strength   * Well established protocol * Proven flexibility * Tolerant of communication outages due to stateless design | Weakness   * HTTP Server offers a single point of failure * Transmitting C-ITS traffic requires design and development work * Transmission delays due to poll (rather than push) model * Due to stateless design, session and message management requires design and development work * Potential scalability issues for hundreds of millions of concurrent clients |
| **Environment** | Opportunity   * Familiar to developers – large talent pool | Threats   * Client-Server model rather than peer-2-peer * Some argue that HTTP is [not suited](https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_http_is_not_enough_for_the_internet_of_things?lang=en) to IoT style environments * Not a C-ITS Standard |

Analysis:

Though difficult to quantify, the amount of design and development work required to offset the weaknesses and threats of this protocol appears to be greater than a number of the other protocols listed in this report. Without further development work, this protocol suffers from many of the same flaws as *TCP Encoded ETSI Message* (above).

The protocol is “stateless”, that is to say, the protocol does not retain a memory of any previous transactions or communications. Additional message and session management may be required to be developed for example to accommodate for powered down vehicle stations.

Additionally, due to similarities between C-ITS and IoT, arguments against HTTP for IoT may be bought to bare against HTTP for C-ITS, such as protocol overhead and potentially, energy efficiency -

“*HTTP is not really ideal for many of its* [IoT] *special needs, such as:*

* *Emitting information from one to many*
* *Listening for events whenever they may happen*
* *Distributing small packets of data in huge volumes*
* *Pushing information over unreliable networks*
* *High sensitivity to*
  + *Volume (cost) of data being transmitted*
  + *Power consumption (battery-powered devices)*
  + *Responsiveness (near real-time delivery of information)*
* *Security and privacy*
* *Scalability*”

As such, though this protocol is extensible and in wide use, it is not recommended for deployment in the pilot.

Ref-

<http://stephendnicholas.com/posts/power-profiling-mqtt-vs-https>

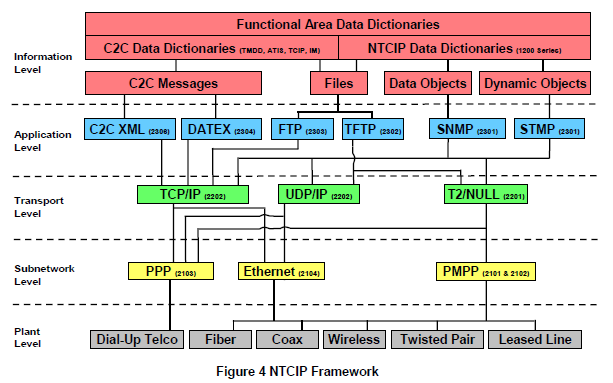
<https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_http_is_not_enough_for_the_internet_of_things?lang=en>

**NTCIP**

The National Transportation Communications for Intelligent Transportation System Protocol began its development in 1993. NTCIP was developed in conjunction with, and is currently in use by the US Department of Transportation and was specifically designed “*to allow electronic traffic control equipment from different manufacturers to operate with each other as a system*” – US DoT.

The protocol consists of approximately 60 documents ranging from naming conventions to testing certification guides. It has its own architecture and data dictionary sets and additionally, appears to have been built to incorporate legacy communications media and protocols such as “Dial-Up Telco”

The following figure represents the protocol choices available in NTCIP and their linkages.



The Harmonisation Task Group 7 web site (<http://htg7.org/>) indicates that SNMP (Simple Network Management Protocol) is planned/draft for use in the USA, however, includes no security measures, meaning that possibly, a Transport Layer Security tunnel would have to be established to protect every connection (ISO 15784-2).

|  |  |  |
| --- | --- | --- |
|  | **Helpful** | **Harmful** |
| **Inherent** | Strength   * Established protocol * Capable of suiting a wide range of traffic solutions due to flexible protocol choice within the framework | Weakness   * Client-Server orientation with little peer-2-peer communication * No mechanism to find other C-ITS devices * Must continue to support legacy equipment and protocols, potentially complicating design work * No media/network selection * Few inbuilt security features (implies use in a closed, trusted network) |
| **Environment** | Opportunity   * Widely used in USA transport jurisdictions | Threats   * Limited use within Australia * Limited visibility outside of the transport industry * Lengthy standards review cycle (5 years approx. - *TransMAX*) |

Analysis:

This protocol does not appear to accommodate for V2V or V2I transmission. Although not a requirement, NTCIP has few (if any) integration points with IoT environments/devices, the integration of which has the potential to further enhance the value and uptake of C-ITS. Additionally, attempting to retrofit ETSI or IEEE C-ITS standards into NTCIP may have limited benefit to the pilot due to NTCIP’s low uptake within Australia.

The establishment and management of millions or hundreds of millions of TLS connections may also prove problematic, or at least expensive to operate and maintain.

SNMP *can* be extended to transfer ETSI defined messages (as could many protocols), and the protocol has been implemented on many computing platforms. However, the protocol was originally designed to manage network devices, not communicate content on road traffic environmental conditions.

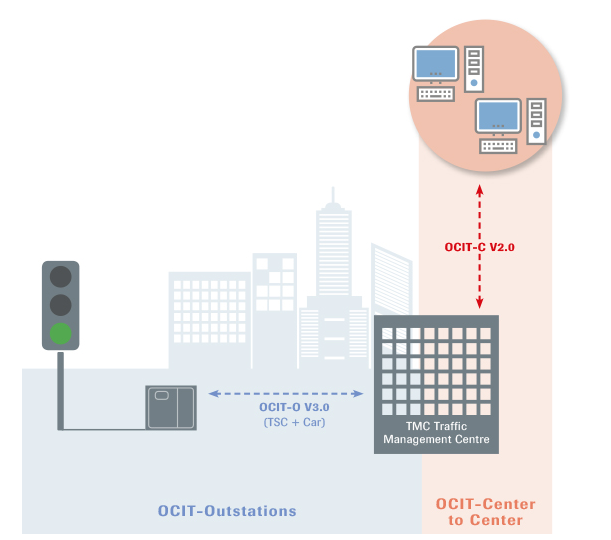
*“NTCIP is intended for fixed-point to fixed-point communications between computers in different systems or different management centers, and between a computer and devices at the roadside. Current NTCIP standards are not intended for use in devices owned by individual travelers; other standards either currently exist or are in development for those purposes.”* - NTCIP 9001 version v04

This protocol is therefore not recommended for the pilot.

Ref: <https://www.ntcip.org/library/documents/pdf/9001v0406r.pdf>

**OCIT-C**

Open Communication Interface for Road Traffic Control Systems - Center to Center (OCIT-C) is a Centre to Centre communications protocol developed in Germany. It was proposed for use by the Eco-AT trial in Europe for centre to field communications. Of note in this trial is the use of OCIT-C rather than OCIT-O. As above OCIT-C was originally designed for (traffic management) centre to centre communications and OCIT-O, for centre to field communications. The reasoning on this design choice is not present in the Eco-AT v4 specifications. The following diagram (from the OCIT website) outlines the protocol’s intended use case -



|  |  |  |
| --- | --- | --- |
|  | **Helpful** | **Harmful** |
| **Inherent** | Strength   * Well established protocol * Tolerant of communication outages | Weakness   * OCIT-C Server offers a single point of failure * Design and development work is required to transmit C-ITS traffic * Communication delays where Client-server model requires vehicle to poll server * Session and message management, design and development work required * Potential scalability and support issues for millions of concurrent clients * No inherent security, potentially designed for a closed network * Not designed for lightweight transfer of information * Client-Server model rather than peer-2-peer |
| **Environment** | Opportunity   * Familiar to ITS engineers – some talent pool * Used in Eco-AT | Threats   * Some argue that HTTP is [not suited](https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_http_is_not_enough_for_the_internet_of_things?lang=en) to IoT style environments |

Analysis

“*Characteristic properties of OCIT-C are:*

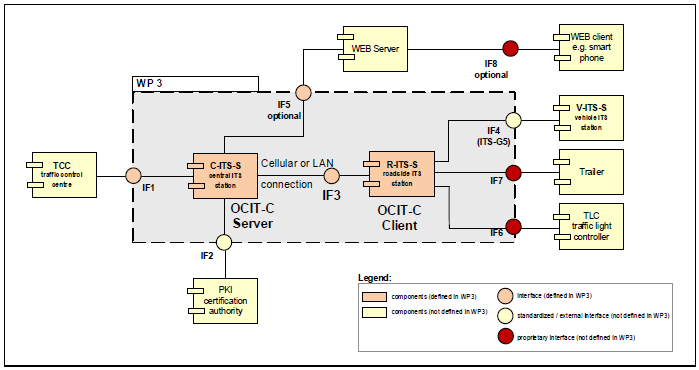
*• exchange protocol based on the standard SOAP with simple request-response communi-cation pattern (direct retrieval of data).*” OCIT-C v1.1

Based on SOAP, many of the features and analysis performed on HTTP may be applicable to OCIT-C.

*“The required bandwidth depends on the amount of clients, object types and objects in the system. Therefore, there is nothing to say here regarding the bandwidth. A local area net-work (LAN) between the central applications will however offer a sufficient transmission ca-pacity.”*

As per HTTP/SOAP, OCIT-C was not specifically designed for low latency, low bandwidth environments and carries a number of overheads that are attempted to be avoided in the C-ITS environment through a number of mechanisms, including UPER encoding of messages. Eco-AT (v4 IF3, section 4.1) have proposed that that an ETSI message by inserted directly into the OCIT-C payload for at least one scenario.

ECo-AT used OCIT-C to send information from the Transport Jurisdiction to the roadside station (not vehicle). As below -



Eco-AT has essentially used a HTTP “long poll” (wait4Get) to emulate a push model down to the roadside station. While this model is sufficient for relatively stable roadside stations, were the TMR CAVI pilot attempt to use this protocol for communications to the vehicle, it is unclear how the model would handle transient communications with a device, and also handle the continual change of the device’s network identity.

Additionally, there is very little security built into OCIT-C, Eco-AT propose that this can be overcome with individual encrypted tunnels from each roadside station back to the Transport jurisdiction (vpn connections). While this is fine for a few thousand stations, scalability has not been proven for millions of devices.

This protocol requires the development of an OCIT-C server and require that each station provider build an OCIT-C client. In a full production sense, the OCIT-C server would have to scale to and maintain millions of concurrent connections. OCIT-C is not well used outside of the European transport management environment, reducing the pool of experienced resources to draw from. This development work and infrastructure management was not seen as desirable by the team.

Unlike OMD-DDS or MQTT, the research undertaken by the team was not able to locate cross-platform OCIT-C server and client implementations (open source or otherwise) within the research timeframe.

Although the Eco-AT implementation of includes a number of adaptable concepts and ideas, the team was of the opinion that it’s planning, design, development, and testing would not be able to completed within the timeframe. The protocol’s focus on transport management may also limit its future integration with non-transport related environments and there are open questions as to its scalability and bandwidth overhead. As such, this protocol is not recommended for research and development by TMR.

**MQTT**

“*MQTT (formerly the MQ Telemetry Transport) is a lightweight protocol that’s primarily designed for connecting power-constrained devices over low-bandwidth networks. Though it existed for over a decade, the advent of M2M (machine to machine communications) and Internet of Things (IoT) has made it a popular protocol.*

*Developers aspiring to build IoT solutions need to learn MQTT, which is quickly becoming the most preferred protocol for connecting devices to the cloud. Enterprise cloud platforms such as Amazon Web Services, Microsoft Azure, and IBM Watson expose their IoT PaaS through MQTT*

*…*

*MQTT was created way back in 1999 by two engineers — Andy Stanford-Clark (IBM) and Arlen Nipper (Eurotech). They had to invent a new protocol for connecting oil pipelines over unreliable, satellite networks.*

*The motivation for designing MQTT was to create a lightweight and bandwidth-efficient protocol that was data agnostic with support for multiple levels of Quality of Service (QoS)*

” - <https://thenewstack.io/mqtt-protocol-iot/>

MQTT is a publish/subscribe style protocol, pushing and pulling all messages through a central message broker. It includes session and message management and is supported by the major cloud platform providers.

MQTT uses “topics” to separate data into categories, which on publication by one client, is relayed to all subscribed clients.

|  |  |  |
| --- | --- | --- |
|  | **Helpful** | **Harmful** |
| **Inherent** | Strength   * Flexible payload able to directly accommodate ETSI communications * Prebuilt Publish/Subscribe framework (supports push rather than poll) * Open source code available * Free, prebuilt clients available for popular platforms * Simple protocol, few primitives * Retained messages and persistent sessions suitable for intermittent connections * Potentially able to be implemented by vendors in under 3 weeks * Built-in message and session management * Includes message QoS mechanism * Changing brokers replicates jurisdiction handover requirement | Weakness   * “Broker” offers a single point of failure * Potentially large broker resource requirements * Rudimentary security features * Potentially too flexible, no guides or data frameworks built in. Data standards and formats must be designed from a zero base and agreed by all clients. |
| **Environment** | Opportunity   * Locally, Ipswich City Council, SmartCities pilot uses MQTT * Supported today by major cloud provider IoT platforms * AWS 80 million device use case * Built for and in use by IoT implementations * Comparatively large number of studies * Has been accepted by both ISO and OASIS standards bodies * Well suited to central authoritative source sending and receiving data to and from multiple clients | Threats   * OMG-DSS proposes a similar concept with far greater peer-2-peer capability * Untested in C-ITS use case * Not in current C-ITS standards set * Perception that IoT environments are insecure |

Analysis

These advantages would enable the delivery of a robust service in a limited timeframe. The use of “topics” to define geographic regions has the potential to solve bandwidth issues across a 3G/4G network (or any shared network). The use of brokers to define jurisdictional boundaries allows jurisdictions to maintain their separation of powers and responsibility.

However, the protocol includes three primary risks –

Reliability – The central broker poses a high resiliency risk to transport jurisdictions.

This risk is able to be significantly reduced with traditional on-site high availability infrastructure, or with the use of Cloud platforms.

Standards based – Though the protocol has been accepted by both ISO and OASIS, the protocol is not currently part of the suite of C-ITS standards and may not move in the same direction of further C-ITS standards or implementations.

As decisions have not been finalised by standards bodies, this risk could be mitigated, potentially to a large degree, through presentation of this use case to standards bodies.

Peer to Peer - The protocol does not adhere to the peer to peer model and concept of C-ITS.

This risk is by far the more difficult to mitigate, however, many of the “centre to field” patterns listed on HGT7.org (for example “In Vehicle Signage”, “Reduced Speed Zone Warning”, etc) are a hub and spoke design and do not included a peer to peer component.

Due to the simple nature of the protocol, its inbuilt message and session management, its potential for compatibility with IoT, client availability and its low time to deployment, MQTT is recommended for the pilot and a candidate for consideration by the larger C-ITS community.

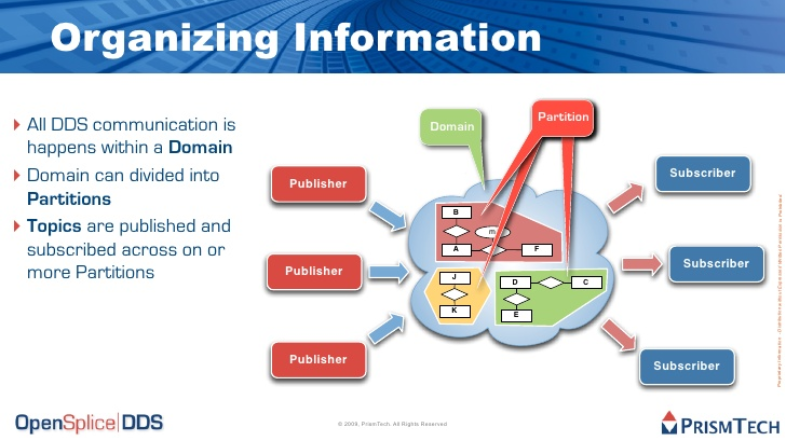
**OMG-DDS**

“*The Data Distribution Service (DDS™) is a middleware protocol and API standard for data-centric connectivity from the Object Management Group® (OMG®).*” - <http://portals.omg.org/dds/what-is-dds-3/>

Similar to MQTT, the protocol uses a publish/subscribe model to distribute data. However, unlike MQTT, all participants may host their own data and could be thought of as publishers, subscribers *and brokers* simultaneously. As such, each C-ITS station must track both its own subscriptions and those clients that are subscribed to it.

OMG-DDS is a peer to peer protocol and claims that it is “uniquely data centric”, that is to say, the protocol both includes the contextual information that the receiver will need to understand the data and abstracts an application from the process of transferring of data. This enables an application programmer to focus on the data rather than the data transfer.

The protocol includes a framework for managing and organising data -



<https://www.slideshare.net/Angelo.Corsaro/omg-dds-the-data-distribution-service-for-realtime-systems>

There are comparatively few peer to peer protocols that follow a purely decentralised approach. Even more popular peer to peer protocols such as BitTorrent, either use “tracker nodes” or “bootstrap nodes”, the purpose of which is to assist in the identification of peers to start participating in the peer to peer network. RTI Connext DDS (an implementation of OMG-DDS) overcomes this issue with an “initial peer list” that is able to be pre-programmed into devices.

<https://community.rti.com/static/documentation/connext-dds/5.3.0/doc/manuals/connext_dds/html_files/RTI_ConnextDDS_CoreLibraries_UsersManual/Content/UsersManual/ConfigPeersListUsed_inDiscov.htm#discovery_507287096_336417>

|  |  |  |
| --- | --- | --- |
|  | **Helpful** | **Harmful** |
| **Inherent** | Strength   * Data is sent with context (Field:value) * Prebuilt Publish/Subscribe framework * Open source code available * Almost pure Peer to peer model – good resilience | Weakness   * More complex protocol requiring a more complex design (as compared to MQTT) * Potentially more difficult to troubleshoot * Potentially larger message payload/greater bandwidth requirements * ‘Initial peer list’ may still offer a single point of failure * Larger ‘client’ runtime resource requirements(for example, connection tracking) than hub/spoke model * Unclear as to how the protocol would handle the C-ITS complexity of rolling pseudonymous identities |
| **Environment** | Opportunity   * In use by US Navy and EuroControl * In use by US Power companies | Threats   * Lack of support by major ICT (Cloud) suppliers * Untested in C-ITS use case * Relatively unknown protocol * Largest OMG-DDS case study features only 1m nodes |

Analysis:

OMG-DDS introduces a number of new terms and concepts into C-ITS over and above a typical publish/subscribe model including – Domains, Domain Participants, Discovery, Quality of Service. Designers of the protocol claim that it has close to near time performance.

Of the protocols examined in this paper, OMG-DDS and ISO/TS 17429 most closely align with the peer to peer concepts inherent in C-ITS.

In theory, because OMG-DDS is a distributed protocol, workload has the potential to be shared amongst each client rather than a central location. However, this is likely to increase client resource requirements.

This protocol receives support from the Object Management Group and has commercial implementations within a small/limited number of organisations. This improves the likelihood that this standard will be maintained and supported.

OMG-DDS has some mention through the harmonisation task group architectural reference site (HGT7.org), however, it appears primarily as its own service under “Data Distribution”.

Under HGT7 use of OMG-DDS *does* appear in a number of centre to field (roadside, not vehicle) communications, however, these use cases tend to imply a hub and spoke communications requirement (with the Transport jurisdiction as the hub) rather than peer to peer and does not appear to be fully taking advantage of OMG-DDS.

In order for C-ITS to take advantage of the peer to peer nature of OMG-DDS, a number of applications, including Transport management applications may need to move from a centralised data storage and processing design to a distributed, loosely coupled design.

Additionally, according to its feature set, OMG-DDS may theoretically have the ability to repeal and replace a number of C-ITS data exchange/communication protocol standards (including ISO/TS 17429, below) developed over the past decade, including low latency, safety critical V2V communications.

Until this occurs it is TMR’s opinion that the advantages offered by OMG-DDS to the pilot remain largely unused, and potentially out weighted by the additional resources required for data planning, research, development and testing of this protocol.

**ISO/TS 17429:2017(E) Intelligent transport systems – Cooperative ITS – ITS station facilities for the transfer of information between stations**

This standard was published in March of 2017. It provides a publish/subscribe facility for C-ITS applications. The standard describes its purpose as –

”*This Technical Specification specifies generic mechanisms enabling the exchange of information between ITS stations for applications related to Intelligent Transport Systems.”*

The standard is integrated with the Communications access for land mobiles (CALM) suite of ISO standards. In ISO 21217:2014 the standard is referred to as *ISO/TS 17429, Intelligent transport systems — Cooperative systems — Profiles for processing and*

*transfer of information between ITS stations for applications related to transport infrastructure management, control and guidance*. This change of name perhaps reflects the standards group’s intension to have the standard apply to generic applications rather than just infrastructure management applications.

The suppliers at the workshop had heard of this standard in most cases, however, no steps had been taken toward its implementation for C-ITS. **At this stage European station suppliers have hard coded communication channels for applications**.

|  |  |  |
| --- | --- | --- |
|  | **Helpful** | **Harmful** |
| **Inherent** | Strength   * Almost pure Peer to peer model – good resilience * Along with the ISO 24102 suite, addresses the “best network link” issue | Weakness   * Complex protocol requiring a more complex design (as compared to MQTT) * Potentially more difficult to troubleshoot * Larger ‘client’ runtime resource requirements(for example, connection tracking) than hub/spoke model * Unclear as to how the protocol would handle the C-ITS complexity of rolling pseudonymous identities * Does not accommodate for multiple jurisdictions * Appears to rely on DSRC for locating C-ITS devices |
| **Environment** | Opportunity   * ISO C-ITS standard implementation | Threats   * Vendors have not implemented this standard * The standard itself is not easy for a technically qualified individual to understand |

Analysis

This standard appears to share many of the traits (and therefore analysis) of OMG-DDS, including –

* A peer to peer approach.
* A publish/subscribe model; and
* Registration of quality of service requirements.

This standard offers one significant advantage over both MQTT and OMG-DDS, and that is it’s integration with *ISO 24102‑6, Intelligent Transport Systems — Communications access for land mobiles (CALM) — ITS station management – Part 6: Flow management*.

*24102-6* allows for the dynamic choice of network media for any particular transfer to occur depending on the environment of the station and the condition of its network links at any given time.

Unfortunately, due to its differences with both MQTT and OMG-DDS and its integration with the ISO C-ITS suite, “off the shelf” products and source code cannot be used, and design, development, implementation and testing would be required by both the Transport jurisdiction and the station providers. Both TMR and the station providers did not believe this development work could be achieved in the specified 3 week period.

The standard has no support or maintenance outside of a C-ITS environment.

Although this standard is part of the ISO C-ITS suite, it is not recommended for immediate deployment in the pilot. However, due to the varied and changing nature of the physical environment C-ITS is intended for, dynamic allocation of communication channels has major benefits for C-ITS. This paper recommends that the integration of this feature be further explored by the pilot.

# Pilot implementation of MQTT

Following this recommendation, the systems integration partnership that was formed under the pilot successfully integrated the Transport jurisdiction’s public facing traffic incident management system (QLDtraffic.qld.gov.au) with vehicle stations from two station providers for the Road Works Warning use case. This was achieved within a 3 week period using a data transformation engine and cloud provisioned MQTT.

It is the vision of TMR to extend and template this solution to support –

* All C-ITS pilot use cases
* GeoTiling within jurisdictions, to reduce network bandwidth overhead, whilst still preserving privacy
* A worldwide, customisable, delegated authority model, to enable jurisdictions to manage their own C-ITS enabled traffic environment and the division of responsibilities within that environment
* The ability to seamlessly transfer ITS-S between jurisdictions
* Monitoring of transport infrastructure
* Interoperability with IoT devices and smart cities initiatives whilst still maintaining C-ITS authenticity, integrity and privacy principals

# Integration with C-ITS

The ability of the three frameworks with the highest strengths, MQTT, 17429 and OMG-DDS to meet the pilots “technical” requirements (See TMR C-ITS pilot requirements above) is summarised in the following table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **MQTT (CAVI)** | **MQTT** | **17429** | **OMG-DDS** |
| Pub/Sub | Y | Y | Y | Y |
| Clearly defined publication structure | Y | N | N | N |
| Flexible publication structure | Y | Y | Y | N |
| Supports persistent messages | Y/N (through workaround) | Y |  |  |
| Payload agnostic | Y | Y | Y | Y |
| Comparative Client Resources | Low | Low | Medium (subscription tracking) | Medium (subscription tracking) |
| Bandwidth | Low (no data context) | Low (no data context) | Low (no data context) | Low/Medium (data travels with context) |
| QoS | Y | Y | Y | Y |
| Network Selection | Y/N (Hardcoded) | N | Y | N |
| Peer to Peer | Y (5.9 only) | N | Y | Y |
| Hub n’ Spoke | Y | Y | Y (subset of Peer to Peer) | Y (subset of Peer to Peer) |
| Global Discovery Process | Not Required | Not required | Not defined | Not defined (though implementations have used an “initial peer list”) |
| ISO or Bespoke | ISO | ISO | ISO | OMG/Bespoke |
| Cloud provider support | Y | Y | N | N |
| Open source community support | Y | Y | N | N |
| C-ITS Standard | N | N | Y | N |

This is not a complete list of features, however, it raises a number of discussion points and begins to highlight both missing and complementary components with regard to station to station communications.

# MQTT Protocol integration with ETSI C-ITS

This section describes how MQTT could aligns or ‘fit’ in with the existing ETSI standards (as at 06/2018).

**Layer Mapping**

The ETSI communications architecture (*ETSI EN 302 665 V1.1.1*) defines the facilities layer as –

“The ITSC facilities layer contains functionality from the OSI application layer, the OSI presentation layer (e.g. ASN.1 encoding and decoding, and encryption) and the OSI session layer”. Under the OSI model, similar to HTTP and SMTP, MQTT is considered an application layer protocol and by this definition most logically sits as a communications choice in the Facilities Layer of the C-ITS architecture.

**Message flow and Security**

The existing ETSI protocols appear to describe a scenario where security functions are built into its GeoNetworking functionality at the Networking & Communications layer. MQTT however, being a protocol that manages messages rather than networking, sits best (as above) at the Facilities layer. This means that for the existing ETSI security mechanisms to be enabled, for example, message signing, the communications path would look something like the following diagram –



The ETSI message (DENM for example, generated at the Facilities layer) would travel from Facilities Layer to Networking, from Networking to Security (to be signed), then passed back from Security to Networking, from Networking back to Facilities (to be delivered by the MQTT message handling), then back from Facilities to Networking (TCP) and finally to the cellular access layer.

While this may be technically possible, it is not defined by the standards and is relatively inefficient. It would be far more efficient (less hops between layers) for the message to be passed directly from the Facilities Layer to the Security Layer before being passed down the communications stack, as follows –



This interface between the Facilities Layer and the Security Layer exists under the standard however, appears to be only loosely defined as follows - *ETSI TS 103 301 V1.1.1 (2016-11) Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services* states -

“*4.4.3 Interface between Facilities layer and Security entity*

*The infrastructure service may exchange information with the Security entity*.”

Unfortunately, the standard does not specify any further details on this interface (SF) and the standard that appears to have been intended to cover its definition - *ETSI TS 102 723-9: "Intelligent Transport Systems; OSI cross-layer topics; Part 9: Interface*

*between security entity and facilities layer"* remains unpublished.

With regard to Infrastructure services provided by the Facilities Layer, the same standard (*ETSI TS 103 301*)also states –

“*4.5.1 Security for messages used by infrastructure*

*The security mechanisms for messages used by the infrastructure service as specified in the present document shall use the message authentication with signatures to be verified at the receiving ITS-S with public keys contained in certificates.*”

So it is clear that these messages were intended to be secured by the Security Layer.

In order to reuse the security features built into the ETSI standards, the facilities layer would ideally call the security layer *prior* to passing the message to the networking layer. One of the very few references to the security services that are made available at the Facilities Layer can be found in *ETSI TS 102 940 V1.3.1 (2018-04) “Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management”*.

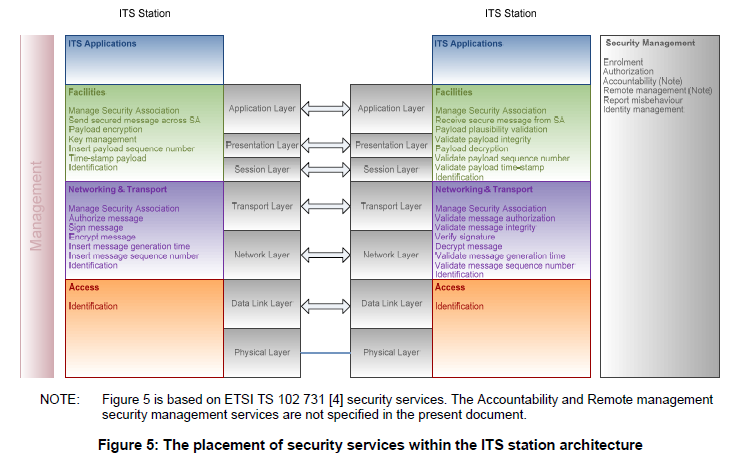


Figure 5 of *ETSI TS 102 940 V1.3.1 (2018-04)* represents the Facilities layer as able to use security services that are associated with encryption, secure session management and plausibility checks, ***without*** being able to call on signature verification or signature generation. This means that any protocol choice other than GeoNetworking will be unable to provide authenticity services without developing a new standard for signing that protocol.

As an example, using TCP as a protocol would require a number of choices in developing the signing algorithm, including the scope of the signature over the packet. This has already been solved for both TCI/IP v4 and v6 using IPSec Null, but would likely require rework for a C-ITS environment.

This rework could be avoided by allowing the Facilities Layer access to the Signing and Verification functions from the Security Layer in the above diagram. This is the same approach that *ETSI TS 103 097 V1.2.1 (2017-10)* implies for DENM messages using the EtsiTs103097Data-Signed data structure and a DENM set as the ToBeSignedDataContent.

In this manner, and as an example, EtsiTs103097Data-Signed data structure could be inserted directly into the payload section of an MQTT publish message at the Facilities Layer, thereby satisfying message integrity requirements.

MQTT as a protocol choice would be made by the station when receiving information from the Traffic Jurisdiction, or communicating with IoT grade devices.

A proposal for the communications parameters of MQTT under ETSI are laid out in the following table

**ISO TS 17423 (2017) – MQTT**

| **Communication Service Parameter** | **ASN.1 type** | **Comment** | **Value** | **Description** |
| --- | --- | --- | --- | --- |
| **CSP\_LogicalChannelType** | LogicalChannelType | Mandatory communication service parameter. | 2, 3, 4 | 2 Service advertisement  channel; 3 Safety of life and property channel; 4 Service channel |
| CSP\_SessionCont | SessionCont | Applicable only for session based flows. | 1023 years, True |  |
| CSP\_AvgADUrate | AvgADUrate | Applicable only for information dissemination flows. | N/A – QoS delivery (1) |  |
| CSP\_FlowType | FlowTypeID | Identifier of a flow type. | <TBD> | MQTT identifier to be assigned |
| CSP\_Max\_Prio | ITSapObPriority | Maximum allowed priority for flows |  |  |
| **CSP\_PortNo** | PortNoInfo | Mandatory communication service parameter, applicable for communication sources and communication sinks. | 1883 |  |
| CSP\_ExpFlowLifetime | ExpFlowLifeTime | Expected lifetime of a flow; not necessarily a  flow related to a session. |  |  |
| Destination communication service parameters |  |  |  |  |
| **CSP\_DestinationType** | DestType | Mandatory communication service parameter. | 0 | Address based type |
| **CSP\_DestinationDomain** | DestDomain | Mandatory communication service parameter. | 4, 8 | DestDomain::=INTEGER{  stationInternal (0),  linkLocal (1),  siteLocal (2),  itsNWlocal (4),  global (8)  } (0..255) |
| CSP\_CommDistance | CommDistance | Applicable only to indicate distance to next neighbour node (outside the ITS-SU). | Global |  |
| CSP\_Directivity | Directivity | Applicable only to indicate communication direction towards the next neighbour node (outside the ITS-SU). | N/A |  |
| Performance communication service parameters |  |  |  |  |
| CSP\_Resilience | Resilience | Any means suited to increase the likelihood of  proper delivery of messages. |  |  |
| CSP\_MinThP | MinThP | Especially meaningful for flows of audio and video streams. Less meaningful for transmission of single messages, e.g. DENM. |  |  |
| CSP\_MaxLat | MaxLat | Especially meaningful for transmission of single messages, e.g. CAM, DENM, SPaT. | 1000 ms |  |
| **CSP\_MaxADU** | MaxADU | Mandatory communication service parameter. Necessary in case of “large ADUs” where due to frame size restrictions in the ITS S access technologies potentially fragmentation is needed. | 128KB | AWS Implementation limitation – in excess of 300MB under the ISO/OASIS standard |
| Security communication service parameters |  |  |  |  |
| CSP\_DataConfidentiality | DataConfidentiality |  |  |  |
| CSP\_DataIntegrity | DataIntegrity |  |  |  |
| CSP\_NonRepudiation | ReqNonrepudiation |  |  |  |
| CSP\_SourceAuthentication | SourceAuthentication |  |  |  |
| Protocol communication service parameter |  |  |  |  |
| CSP\_Protocol | ProtocolReq | A complete registered non-parameterized  protocol stack | TCP | Relies on session based communications channels |
| CSP\_SpecificCommsProts | SpecCommsProts | List of specific non-parameterized  communications protocols |  |  |